Management Challenges in the Design Phase of Collaborative R&D Projects

by

Nicholas Taylor Mollison, B.S.M.E.

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

August 2009

Management Challenges in the Design Phase of Collaborative R&D Projects

Approved by
Supervising Committee:

Anthony P. Ambler

Joseph H. Beno

Abstract

Management Challenges in the Design Phase of Collaborative R&D Projects

Nicholas Taylor Mollison, M.S.E.
The University of Texas at Austin, 2009

Supervisor: Anthony P. Ambler

The engineering and design of complex systems often requires that multiple design tasks be executed in parallel or overlapping efforts. When the design of individual subsystems is distributed among multiple organizations, challenges arise with respect to managing design productivity and coordinating successful collaborative exchanges. Research and development engineering projects compound these challenges further due to their inherently greater uncertainty. This report examines several factors that influence design productivity in the collaborative research and development environment, including the selection of subsystem interfaces, design information management, and complexity management. A collaborative research and development project to upgrade the Hobby-Eberly Telescope is introduced to provide case examples and illustrate the proposed value of subsequent management recommendations.

Table of Contents

List of Tables

List of Figures

List of Acronyms

Term:	Definition:	First Use:
CAD	Computer Aided Design	pg. 2
CCPM	Critical Chain Project Management	pg. 41
CEM	Center for Electromechanics	pg. 10
DSM	Dependency Structure Matrix	pg. 37
FMEA	Failure Modes and Effects Analysis	pg. 27
FTP	File Transfer Protocol	pg. 26
HET	Hobby-Eberly Telescope	pg. 2
HETDEX	Hobby-Eberly Telescope Dark Energy eXperiment	pg. 2
ICD	Interface Control Document	pg. 12
MDM	Multi-Domain Matrix	pg. 40
IGES	Initial Graphics Exchange Specification	pg. 26
MDO	McDonald Observatory	pg. 10
PC	Personal Computer	pg. 6
PERT	Program Evaluation and Review Technique	pg. 41
PDM	Product Data Management	pg. 25
PFIP	Prime Focus Instrument Platform	pg. 10
R&D	Research and Development	pg. 3
RAM	Random Access Memory (computer memory circuit board)	pg. 6
SALT	Southern African Large Telescope	pg. 10
STEP	Standard for the Exchange of Product model data	pg. 26
WFC	Wide Field Corrector	pg. 9

1. Overview

1.1. Introduction

Collaborative, concurrent, simultaneous, or parallel engineering...regardless of the specific nomenclature used, collaborative engineering is becoming increasingly more prevalent and important to engineering projects that include a research and development component. For any technical manager involved with the design of products or processes, directing a collaborative design effort will pose difficult challenges. It is vital for design managers to have a sound grasp on several issues specific to the collaborative environment, as societal trends will only continue to increase the scope and necessity of inter and intra-organization collaboration in fields that require routine technological advancement. Such trends include the need for professional and technical specialists in high-tech sectors, industrial globalization, time-to-market economics, and modular and system-based product architectures [15], [17]. A final trend worth individual attention is the widespread sharing of digital information and the rapid development of internet communication tools and software. The ability to transmit file data, audio, and video communication in near real-time via high-speed internet connection has enabled design collaboration to span cities, nations, and continents. Aerospace, consumer electronics, and automobile manufacturers are just a few examples of participants in truly global engineering collaborations.

This report focuses on strategies for understanding and managing design effort productivity in increasingly complex collaborative environments which are accompanied by equally complex engineering objectives. Case examples from the Hobby-Eberly Telescope "Dark Energy eXperiment" project will be introduced to illustrate several facets of a recent collaborative design project involving significant engineering research and development.

1.2. BACKGROUND

The Hobby Eberly Telescope (HET) is the third largest operational ground-based telescope in the world at the time of this writing [8], (Figure 1). Commissioned and managed by a consortium of five universities, the HET resides at the University of Texas McDonald Observatory near Ft. Davis, Texas. The "Wide Field Upgrade" to the Hobby Eberly Telescope, which will enable the Dark Energy eXperiment (together referred to as HETDEX), is a an estimated $33 million upgrade to the HET's optics, instrumentation, electromechanical positioning system, and electronic controls for the purpose of upgrading its capabilities to enable precise measurements that will contribute to scientific understanding of dark energy[1] [2]. Within the present context of this report, the project has reached mid-term between its engineering kick-off and scheduled commissioning date.

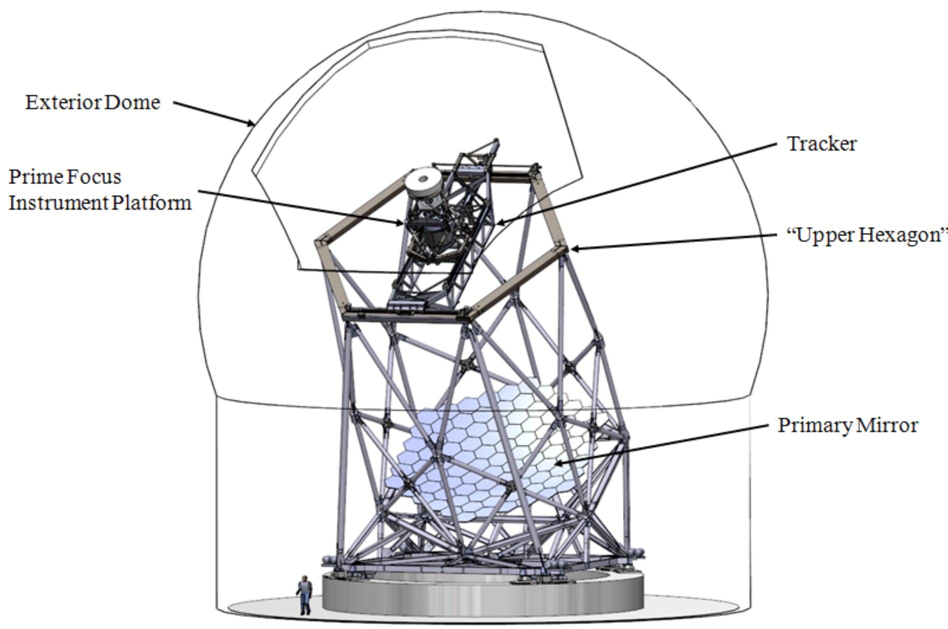

Figure 1: The Hobby Eberly Telescope (CAD model created by HETDEX team).

[1] "Dark energy" refers to the phenomenon thought to be behind the Universe's increasing rate of expansion, which has been measured but not yet fully explained by the scientific research community. HETDEX will perform measurements of the expansion history of the Universe, which may contribute to a more complete scientific theory of dark energy [10].

There are some aspects of the HETDEX upgrade project that will not directly compare to industrial R&D projects. The greatest of which being that it is a one-time engineering effort which will not be repeated by the same group of designers in subsequent projects, nor are there considerations for sales or production in quantity beyond first-prototype hardware. As a result, there are also fewer engineering compromises between performance and cost, and other similar tradeoffs often encountered in the realm of industrial or commercial product development.

Barring these differences, the design phase itself has many characteristics that share familiarity across R&D projects, irrespective of discipline or industry. The scope of the HETDEX upgrade requires engineering expertise from several disciplines, and from groups that are dispersed among separate organizations and, in some cases, spread geographically around the globe. There were many design requirements and constraints either still evolving or yet to be determined at the onset of the project (some of which could not be determined prior to initial concept development). Time-to-market pressure has also been present in the form of completing the dark energy experiment in time to maximize scientific relevance, and to contribute the project's engineering achievements to benefit similar projects within the scientific community. Finally, despite the one-time nature of the project, significant design consideration has been granted to reliability, maintainability, and manufacturability of the various components.

To satisfy all design objectives within the environment described above, it can be inferred that a substantial degree of collaboration is required among design participants. Accounts of noteworthy interactions pertaining to the parallel efforts of the parties involved will be given in subsequent sections of this report.

1.3. THE CHALLENGES OF MANAGING COLLABORATIVE DESIGN

Developing and engineering a system as complex as the one required for the HETDEX upgrade, and within a span of less than four years from start through production, testing, installation, and commissioning, is only one factor driving the need for parallel design efforts. Tightly integrated subsystem functions and interfaces have made parallel and collaborative design a necessity due to the “chicken and egg” problems of large-scale system design. There are many instances where one subsystem cannot be designed without requisite knowledge of the other, nor can one design be substantially altered without reflexively impacting another.

Both the goal of design managers, and the obstacle to reaching this goal posed by the parallel collaborative environment, can be stated in very fundamental terms. Goal: Maintain a high level of design productivity by enabling high-quality designs and limiting unnecessary design iteration. Obstacle: Design decisions must be made with less than optimal quantity and quality of information. Even though the problem statement appears fundamental, there exists no single easy solution, set of tools, or list of rules that can address its full implications. Some say that the application of “best practices” is the most that can be hoped for in lieu of a guaranteed design process [1], while others contend that there are no best practices and instead an overarching “way of thinking” is required [16]. The resounding theme, though, is that design management must be as dynamic, adaptable, and change-tolerant as the component designs themselves to make collaboration a success. Within the collaborative environment, a compromise must be made between attempting to resolve uncertainty and accepting that collaborative design will be an organic and uncertain process. Design management may then consist of making clear and decisive choices without an overly-inhibited conscience, achieved by collecting the best information available regarding the range of potential consequences and processing that information effectively. The purpose of this report is not to introduce new dogma for best practices or holistic ways of thinking. Instead it will concentrate on illuminating a few key factors that have far-reaching impacts on the collaborative design project outcome, followed by recommendations to assist their monitor and control. These factors include the choices made when dividing a system into its subsystems to form the product architecture, the exchange of design information, and problems that arise from growing complexity.

2. The Implications of System Architecture

"Some problems are so complex that you have to be highly intelligent and well informed just to be undecided about them." - Laurence J. Peter (from [3])

The above quote illustrates the fundamental problem encountered when the output of a design effort will be a system that possesses higher-order complexity as a byproduct. The problem is compounded even further when collaboration is involved, because the "well informed" component becomes that much more difficult to achieve due to additional constraints on communication. System architecture choices may not be immediately associated with communication needs and constraints, but it will be shown that architecture is a deterministic factor.

System architecture refers to the structural arrangement of the system being designed and, more specifically, to the division of the system into subsystems and the resulting connectivity and interactions between subsystems [12], [16]. Dividing a system into functional blocks, subsystems, and even further into modules is what permits parallel engineering to occur, such that the work can be divided amongst various groups who will collaborate to shepherd the modules through the design process simultaneously. The division of work responsibility, however, is not necessarily the underlying motivator for collaboration. Collaboration is required by the fact that each of the subsystems and components that form the product architecture must share compatibility at their interfaces for the system to fulfill its intended function. This section of the report will illustrate how system architecture contributes directly to technical and social complexity, and therefore management complexity in the end.

2.1. System Architecture and Complexity

Each division of the system or product creates at least one interface between system components. In order for the system to work, the components must share at least one type of compatibility with respect to their intended interaction; often in the form of mechanical interfaces, electrical interconnects, communication protocol, or data format, to give some examples. The creation of interfaces has the effect of creating locations of concentrated information within the system [18].

Consider a video adapter card that plugs into the motherboard of a Personal Computer (PC). The video card design has only a few external constraints with respect to the precise dimensions of the circuit board, chip arrangement, chip type to be used, internal electrical connections, etc. This leaves the video card's designer a substantial amount of latitude to make decisions without affecting other aspects of the computer's design. However, at the interface between the video card and motherboard, there must be strict adherence to pre-established parameters such as the geometric dimensions and tolerances of the connector, electrical pin-out arrangement, power and voltage supply requirements, communication protocol, and so on. The designers of the motherboard and video card must each have detailed knowledge of this interface. Furthermore, they must be able to clearly communicate the relevant interface design information to one another, notify each other of proposed changes, reach consensus approval, and transmit design feedback once the changes have been made. In this two-part system, the exchange between designers is relatively straightforward and manageable.

Now consider what would happen in a situation where the motherboard designer discovers that a non-standard design layout will have to be used, such as mounting the video card vertically rather than the usual practice of mounting it horizontally within the PC case. This creates additional, perhaps even unintended interfaces (i.e. interferences) for the video card designer, which in turn requires him or her to have more knowledge of the overall system and to potentially place additional constraints on the design. Now the video card designer may need to know the location and dimensions of the computer's memory circuit boards (RAM) to check for interferences, and the length of the wire leads coming from the power supply to ensure they will still reach the video card. After exchanging emails and telephone calls with the other PC design team members responsible for the motherboard layout, RAM, and power supply designs, the responses may come back: "The RAM layout hasn't been finalized yet, and how long do the power leads to the video card actually need to be?" In reality, if this were also a new and untested component configuration, the involvement of the entire PC team in evaluating this single design change might be even greater in scope. They may have to assess issues pertaining the enclosure design and impact on thermal performance, which could lead to even further design iterations to circuit board layouts and hardware configurations.

The preceding example was constructed to highlight the technical and social complexities that arise in collaborative system design. In the technical domain, changing one interface within a sufficiently complex system had the effect of altering or creating new interfaces, and also created the need for additional component design information and system level knowledge from all designers involved. Within the social domain, the designers also had to become aware of who possessed the relevant information and initiate communications in order to access it. The following excerpt provides a formal conclusion to these observations:

> The value of a system lies in its interfaces, not in the individual components. These interfaces also determine the complexity of the system, because the number of interfaces and interactions between elements grows exponentially with the number of components in the system (Reinertsen, [16]).

The second implication from the PC example is the effect that design uncertainty has on the design team. Without being certain where the RAM will reside on the motherboard, it is also uncertain where the final placement of the video card will be, and therefore the video card designer does not have enough information to accurately specify how long the power leads should be. This leaves two options: Specify enough length to accommodate all locations that are currently being considered, or choose an arbitrary length and revise the specification once the exact location is known. The designer must make the decision with less than complete information, and each option will introduce some degree of iteration into the design process and communications between team members. This leads to the next conclusion: "The variability of the system will approach the variability of its most variable component" (Reinertsen, [16]). In the context of the example, the least certain design aspect introduces an equal amount of uncertainty into at least one other subsystem design variable when it transcends an interface.

In the PC design example, a scenario was presented where a change to one component of the system precipitated a change to another component, which precipitated a change to another, and so on. This type of "change propagation" is not an unfamiliar concept in the realm of engineering design, and it occurs when component changes manifest at an interface [1]. Less often considered is how component change propagation ripples outward from the designer in terms of their information needs, decision making, and team interaction. This can be viewed as change propagation across the "human interface" that exists in collaborative projects. Interactions between designers represent the second-order system in the collaborative design process. Both the technical and the social systems involved can be thought of as dynamic, since disturbances to each will energize some form of system response. Technical and social complexity are each a product of the number of interfaces in the system architecture and in the collaborative organization. Increasing complexity will have the effect of making each system's response less clear and predictable before it is set into motion.

2.2. HETDEX System Architecture Example

The single-most involved engineering effort within the HETDEX upgrade is the replacement of the Hobby Eberly Telescope's "Tracker" system (Figure 2). The Tracker is an electromechanical system that positions the telescope's secondary optics[2] with micron-level precision, so that incoming light can be received by the telescope's many scientific instruments. The Tracker receives its name from its primary function of dynamically tracking along a moving trajectory such that it follows astronomical objects

[2] The HET design utilizes a single, large (9.8 meters wide) reflecting mirror called the 'primary mirror' to gather distant starlight. 'Secondary optics' refers to a second set of smaller focusing mirrors (called the Wide Field Corrector) located on the Tracker and above the primary mirror, which focus the reflected light so that it can be analyzed with scientific instrumentation [2], [10].

as they move across the night sky. To achieve such extreme dynamic accuracy requires nearly a dozen actuators, each with sophisticated sensors and electronic controls.

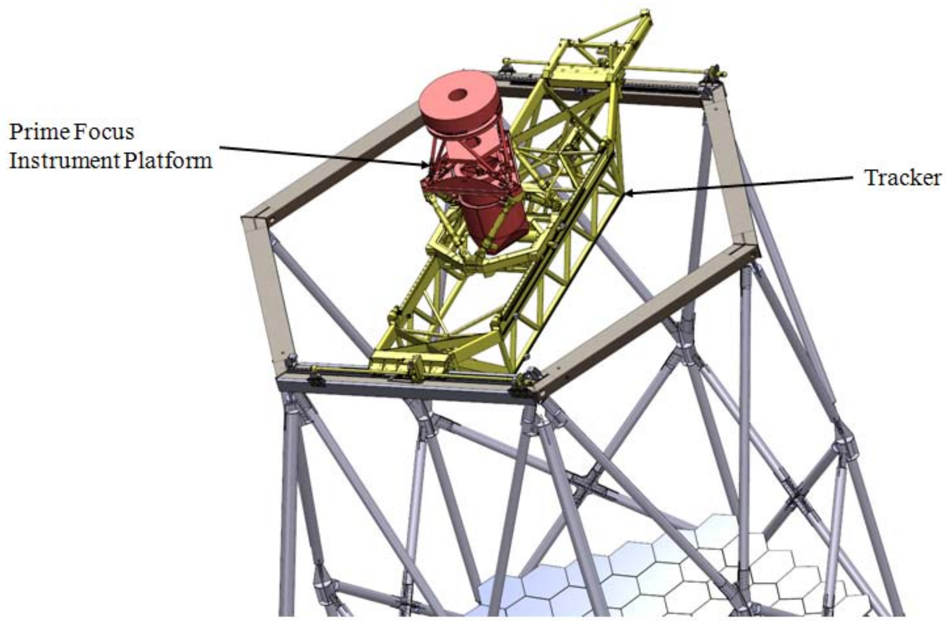

Figure 2: The Hobby-Eberly Telescope's Tracker system (yellow) and Prime Focus Instrument Platform (red) (CAD content created by HETDEX team).

The Tracker upgrade is necessitated by two primary needs of the HETDEX experiment. The first is the replacement of the HET's secondary optics with a new set, housed in an assembly called the Wide Field Corrector (WFC), which will create a wider field of view on the sky [2], [10]. The new WFC, and the instruments that it will feed with starlight, weigh substantially more than the previous Tracker could accommodate. The second HETDEX need is a scientific requirement for the astronomical observation of dark energy, which demands that the telescope perform a greater number of observations each night [2], [10]. Therefore, the Tracker must be able to move back into position following an observation in order to "reset" its viewing orientation on the sky much more quickly. To summarize, the upgraded Tracker will be able to carry roughly twice the payload mass and move more than twice as quickly as the previous Tracker, while also improving positioning accuracy and observing performance.

The Tracker is divided into several electronic and mechanical or opto-mechanical subsystems, some of which are stand-alone and several of which overlap in terms of their function or integration. The design of each subsystem and its individual components is the responsibility of an organization or individual within an organization (Figure 3). The McDonald Observatory (MDO) team, under the astronomy department of the College of Natural Sciences at the University of Texas at Austin, is the organization leading the project. MDO also retains responsibility for the design of several instrumentation components, telescope controls, site modifications, and components within the Prime Focus Instrument Platform[3] (PFIP). They have contracted the Center for Electromechanics (CEM) at the University of Texas at Austin to undertake the design and manufacture of the new Tracker system, and the University of Arizona to perform the same for the Wide Field Corrector. Further subdivision among organizations occurs within the Tracker at the "hexapod,"[4] which is subcontracted to ADS International of Lecco, Italy. Other collaborations within the project include, but are not limited to: Texas A&M University, Astrophysikalisches Institut Potsdam of Germany, and members of the Southern African Large Telescope (SALT) staff.

[3] Prime Focus Instrument Platform (PFIP) refers to the subassembly consisting of of the Wide Field Corrector (which includes the secondary optics), the set of tertiary optics, and the set of sub-structures which join them.

[4] A hexapod is a movable platform supported by six extendable actuator struts arranged in a truss configuration.

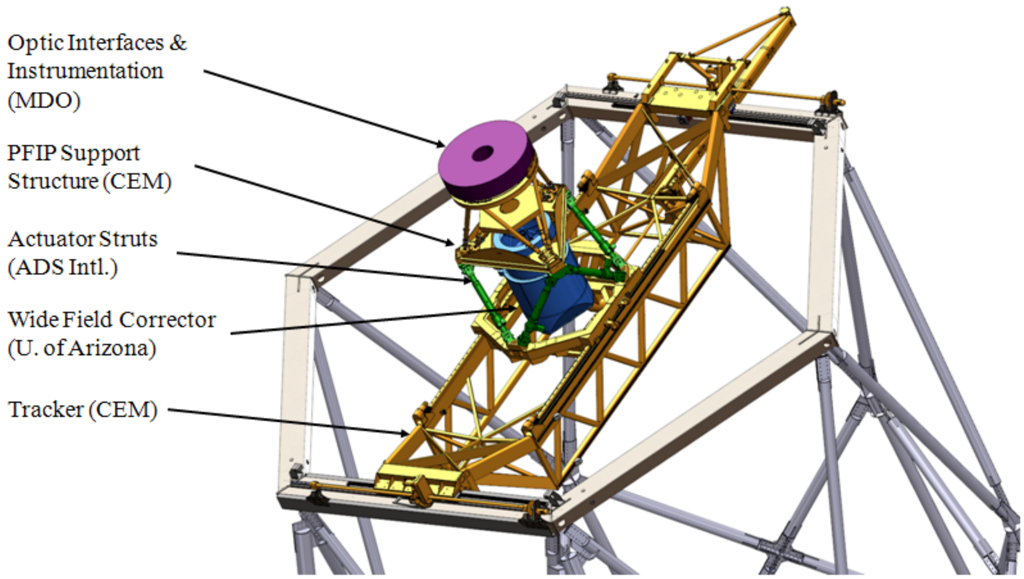

Figure 3: Division of major subsystems and interfaces onboard the Tracker by organization for the HETDEX upgrade (CAD content created by HETDEX team).

Figure 4 provides a schematic representation of the interfaces contained within the Tracker. The partitioning of the upgraded Tracker system is a product of several design influences. This new version of the Tracker will effectively be the "third-generation" of the Tracker device [2]. The first was designed and constructed for the HET as it entered service in the late 1990's. The second was produced for SALT, a cousin of the HET which shares its functional design, but incorporates many component design improvements enabled by the experience gained operating HET throughout the time separating the two projects.

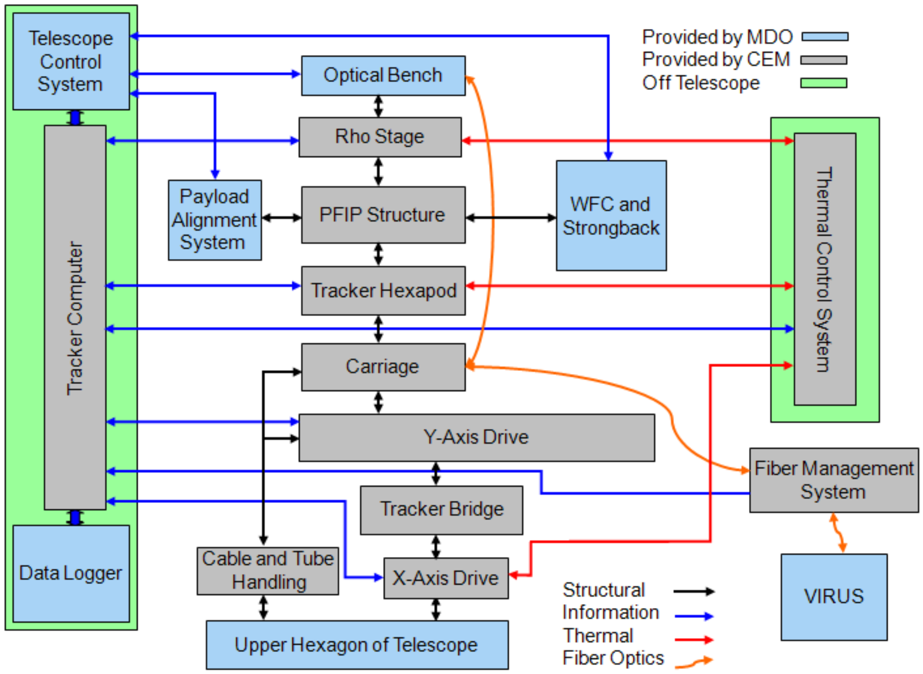

Figure 4: Schematic diagram of the HET's major subsystem interfaces (from [9]).

Many of the Tracker's critical interfaces became well-defined over the course of engineering and designing the two predecessors. For example, the functional interfaces between the Tracker's main structural member (the Tracker Bridge) and its positioning servo-drives are common between all three examples. Other interfaces reflect the substantial differences in specific domain knowledge required to engineer the components of certain subsystems. The interface between the Wide Field Corrector and the support structure linking it to the Tracker is one example (refer to Figure 3). CEM specializes in the electromechanical actuation and control engineering required for the design of the Tracker, which positions the Wide Field Corrector. The WFC is contracted to the University of Arizona for their expertise designing and manufacturing large optical assemblies for ground and space-borne telescopes. In the instances where major interfaces were known beforehand, and for those interfaces that link subsystems under the charge of organizationally or geographically separated groups (therefore requiring extensive collaboration), an Interface Control Document (ICD) was created or identified in the earliest stages of the project. Table 1 lists the major ICD's, and exemplifies the variety of mechanical, electrical, thermal, pneumatic, and data interfaces that are typical of highly automated systems.

Table 1: List of current Tracker Interface Control Documents (from [9]). Each document, which is subject to engineering revision controls, contains a full description of the interface and lists the associated drawings and specifications.

Document #	Interface Described
HX0039-01-01	Upper Hexagon of Telescope Structure to Tracker Bridge
HX0040-01-01	Instrumentation Electronics Mounted to Strongback
HX0041-01-01	Hexapod Actuator Mounts to Strongback and Lower Hexapod Frame
HX0042-01-01	Wide Field Corrector Mount to Strongback
HX0043-01-01	Pupil Assembly Platform to Pupil Assembly Instruments
HX0044-01-01	Rho Stage to Focal Plane Assembly Substructure
HX0045-01-01	Tracker Computer to Telescope Computer System
HX0046-01-01	Tracker Electrical Interfaces and Routing Control Document
HX0047-01-01	Tracker Pneumatic Interfaces and Routing Control Document
HX0048-01-01	Tracker Thermal Management System Interfaces and Routing Control Document

Structural complexity generated by the system architecture is relatively easy to manage with respect to the well defined and pre-established interfaces that separate the major sub-functions of the Tracker. However, a clear picture of architectural implications becomes more difficult to sustain as interfaces penetrate deeper into subsystems and finally reach the component level. At lower levels of subsystem design, more latitude for interface choices exists, and likewise so does uncertainty with respect to their precise functional definition and in-process change control as designs evolve. Figure 3 illustrates the major subsystems, interfaces, and parties responsible for the design of each, but the list of collaborating stakeholders and respective interests that contribute to structural and social complexity does not stop there. Research and development projects, and especially those that are marked by a measure of rarity in their scope and application, often seek to maximize their "one-time" value. What is meant by this statement, is that tremendous efforts will be made to incorporate the benefits of direct and indirect experiences; and to incorporate as many technological developments and practices developed in other similar projects as possible so that the "rare" project will surpass those that came before it to become the new benchmark design. This becomes especially important when the project involves a university's unique scientific research asset like the Hobby Eberly Telescope. Therefore, the engineering objectives for the Tracker stretch beyond upgrading its performance and load capacity. Stakeholders include HET and McDonald Observatory oversight boards, university consortia, and more directly involved participants such as astronomical research scientists, faculty, HET site operations personnel, and technical staff. Significant communication and design input is required from these groups to maximize the future reliability, maintainability, and ultimately the long-term research value that the HET will provide for at least the next twenty years.

In practical terms, exercising design freedom within subsystems and facilitating the interests of a large number of constituents can create many additional interfaces which may be introduced at virtually any phase of the design process. Consequently, these ancillary interfaces are subjected to much less analysis, formal definition, documentation, and change control scrutiny than higher level system interfaces. Yet, the effect they may have both individually and cumulatively on the overall system design can still be measurable due to their introduction of new design constraints and dependencies, thereby expanding the system's overall complexity.

2.3. MANAGEMENT IMPLICATIONS OF SYSTEM ARCHITECTURE

Collaborative engagements contribute to the complexity and uncertainty of the design process, so why undertake them? As complexity increases, so does the designer's need for information and thus communication. The design team itself becomes a complex system with its own requirements, dependencies, and objectives. With increasing complexity, transparency into the consequences of decisions decreases, as does the predictability of decision outcomes [4], [12]. Yet, collaboration and uncertainty are necessary to the design process, especially for enabling innovation in the research and development environment ([10], [16], and [17]). Well-defined design problems that have predictable task progressions and unambiguous solutions rarely generate novel products [16]. Designing a new system for the first time often requires preliminary development of subsystem components in parallel just to generate the contextual information required to formulate an accurate design problem statement. Finally, organizational logistics dictate that systems and subsystems must be separated at established interfaces to suit the availability of resources, and the appropriate functional disciplines and domain knowledge areas of the designers.

The collaborative team environment and the system being designed together comprise a multi-domain system. The diagram constructed by Weber (as presented in [12]) shown in Figure 5 depicts these as the "product and/or system domain" and the "process domain." The product domain refers to technically-oriented attributes of the product or system being designed, and the process domain refers to human resource, organization, and planning aspects of the design process.

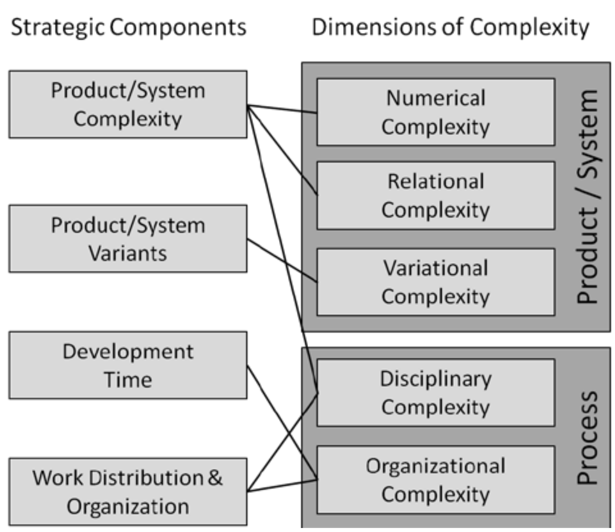

Figure 5: Complexity represented in the system domain and process domain (reproduced from [12]).

When discussing industrial or consumer products, the product domain and process domain may exist independently or exhibit different relationships throughout the various stages of the product lifecycle. When the application of this model is narrowed to the design phase, I would make the assertion that the product domain exists co-dependently and within the process domain. The two are subsequently referred to as the technical domain and social domain throughout this report, in order to add specificity to this model for the context of collaborative design. The social domain was earlier referred to as the "second-order system." Elaborating on this analogy to system dynamics, the technical domain would represent a first-order system. If the technical domain could be isolated, a change to a component design would represent a single step change in the state of the system. When the technical and social domains are coupled, a change to a component design can incite changes from other designers and introduce design iterations, similar to the oscillating behavior of a second-order dynamic system.

The critical item for managers to understand is the nature of the dynamics and uncertainty that are at work in the design process. They cannot be completely isolated or eliminated. Some information cannot be produced prior to the time at which its need it will become critical; and likewise some design approach decisions cannot be finalized without first attempting a baseline solution, despite the inevitability of rework. In the frame of dynamic systems, this would be equivalent to stating that it is not always possible to predict how every element in a system is going to respond; not without first giving the system a small push to see how its interactions unfold. This becomes increasingly true with the uncertainty of early-phase R&D and as system elements and collaborators grow in number. The overarching management goals in the collaborative design process then become to 1) minimize iteration, and 2) understand the sources of system complexity and advantageously manipulate them whenever possible. Achieving these goals can prevent designers from becoming overwhelmed in the collaborative design process and maintain higher levels of productivity [15]. The mechanism behind iteration is design information transfer, i.e. design information is the link or "spring" between two designers about which they will oscillate. Complexity arises from the information dependency relationships formed by the system structure and interfaces, and determines where and how the “springs” will be connected.   The next two chapters of this report concentrate on understanding these two components in practical application to collaborative design, so that management goals can be realized.

3. Information Management

The design development cycle consumes project information and resources and processes them to create the product of research and development engineering, which is all the design information necessary to realize the system and product [16]. Information inputs include requirements, specifications, budget, schedule, and stakeholder influences. Examples of information outputs include design documentation, project cost information, and project status information. A vast amount of information is required to formulate and deliver a complete, engineered design. It was shown in the previous section that, in the context of a system design, this quantity of information is directly proportional to the number of interfaces and interfaces are exponentially proportional to the number of components [16].

In a collaborative project, barriers and impediments to effective communication can leave designers starving for relevant information, or similarly inundated with less consequential design objectives and constraints. This is precisely why collaborative design projects are exceedingly more management intensive at all levels of responsibility, versus non-collaborative projects that predominantly give right of way to the technical issues [7]. This section of the report provides insight into the impacts uncertainty and complexity may have with respect to information and communication during the design phase of a collaborative research and development project.

3.1. Instrumentality of Information to the Design Process

Partitioning a system into subsystems creates interfaces, and interfaces create the need for communication when ownership of subsystems and components is divided at the interface [16], [15]. It has also been shown that external interfaces are locations of concentrated design information, and complex interactions in sufficiently large systems require even greater information about the other subsystems. This is only one facet of the designer's information needs when designing a new subsystem component. Apart from the subsystem's external and internal interfaces, a substantial amount of information is required simply to engineer the component itself. Functional requirements, material properties, stress calculations, usage scenarios, and manufacturing cost estimates are just a few of the many types of detailed information an engineer or designer will use to formulate and refine a mechanical component design. Additional time and effort will be dedicated to producing information output such as presentations for design reviews, design calculation records, prints for manufacture, inspection and acceptance criteria, user's manual and assembly instructions. As a practicing design engineer, I share the perception that the above obligations, which are seemingly tangent to the design process, tend to dominate over the effort expended "designing things." How significant is this really? Management research studies involving practicing engineers, including those of a large auto company, support claims that engineers spend between 8% and 10% of their time performing actual design tasks [6]. Studies have also yielded somewhat conflicting, but no less illuminating data on what designers are doing with the other 90% of their remaining time:

> Getting the right information takes a lot of time during the design process. Statistical investigations with designers in practice report that it takes approximately 25-60% of their working time [7].
>
> It has also been shown that engineering designers spend as much as 30% of their time searching for and accessing information. To try and reduce this "non-productive" time engineering designers tend to use the information that they already possess [6].

The closing statement of the latter quotation implies that, when confronted with an absence of required design information, engineering designers will default to experience or information already in their possession. The conclusion left to be inferred is that this is likely to be sub-optimal information which is being used in the formation of a design solution. In a design scenario that involves a high level of complexity, such behavior is extrapolated to satisficing or, in another form, what has been referred to as an attempt at "taming" the problem [3]. An example of a taming behavior would be restating or rationalizing a difficult problem such that it appears more like a problem which has already been solved, regardless of whether or not the juxtaposition is truly valid [3]. Studies involving practicing engineers seem to support this observation.

> Studies performed by Stauffer et al have shown that often their own [engineer's] decisions are based on rules of thumb, personal preferences and even conjectures, which are formed when there is not enough information to know things with certainty, but enough to make an educated guess [6].

Before exploring the effects that this "default" response to insufficient information has in further detail, it is worth first looking at data that has been gathered regarding the design engineer's preferred sources of information.

> In a survey of over 200 practicing engineering designers, reported that when starting a new task the following information sources were rated as Very Important:
> 60% Colleagues
> 34% Personal Contacts
> 84% Personal Experience
> 17% Representatives
> 12% Consultancy [6]

Interpersonal communication is also cited as being highly important when a high level of uncertainty is believed to exist [6].

If the preferred source of information in situations involving uncertainty is, in fact, person-to-person communication, this would appear to be a benefit for collaborative efforts, and especially those involving research and development engineering where substantial uncertainty exists. Understanding from the very beginning that the project will depend heavily upon collaboration has the effect of forcing open the windows of communication at a much earlier phase in the project [17], [15]. These “meetings of the minds” are essential prior to beginning the actual work, as there is a much greater need to express and clarify each organization’s role, objectives, and expectations in order for the collaborative team to parcel the work, yet still remain cohesive over the long-term [15], [17]. Unfortunately, collaborating across organizations, and especially in teams comprised by a large number of individuals, becomes more difficult once each group becomes embroiled in their respective engineering design tasks. Conflicting meeting schedules, travel schedules, missed connections, and pure willingness to accommodate unscheduled interruptions all become factors that contribute to lessened frequency and actual time duration spent collaborating. The early paradigm of mutual exchanges begins to shift to one of placating the immediate needs of the information seeker versus the immediate preoccupation of the responder. The fact that there are far fewer true collaborations than there are “transactions” at the mid-complete phase of a collaborative design project is a simple and understandable product of everyone’s desire to maintain individual progress once the mutual objective is understood and the tasks have been defined. It is immensely important, however, that the significance of each collaboration does not become underestimated – no matter where each team may be in their respective trajectories or how ever sparse their interactions might be. A study conducted by Frankenberger and Badke-Schaub [7] exemplifies the reasoning behind this assertion:

> In spite of the fact that the designers worked for an average of 80% of their time individually, the importance of the group-related factors in “critical situations” becomes clear by the fact that 88% of the “critical situations” took place in collaborative work of the designers [7].

The “critical situations” referred to within this context were defined as “...where the design process takes a new direction on a conceptual or embodiment design level” [7]. Part of the reason for this apparently imbalanced weight of collaborative outcomes is that individuals contributing to a shared effort typically feel greater impetus to obtain consensus approval and acceptance for their ideas before making final commitment; versus when confronted by similar commitments when working in isolation [7]. The observation taken from the study referenced above has tremendous implications. It suggests a measure of support for the earlier claim that the more frequent “transactional” transfers of information are initiated primarily to enable one’s continuance of work already underway; whereas truly collaborative exchanges most frequently produce, and are sometimes required to formulate decisions which are perceived as being critical to the path of the design process.

Transactional information transfers may not be frequently attributed to pinpoint defining moments of the design process, but their cumulative effect should not be neglected either. Returning to the implications of the “default response” of engineers, i.e. personal experience, rules, of thumb, and conjecture when adequate information is lacking, it can be seen that transactions are an essential component of collaborative design. The same study conducted by Frankenberger and Badke-Schaub was also aimed at identifying external factors and design decision influences that directly affect design process outcomes like cost and quality. “Factors and influences” were narrowed to such things as quality of solution analysis and solution decisions, quality of leadership, time pressure, and information availability. What they found was that “quality of solution decisions” was the single most influential factor to positively or negatively influence overall design cost and quality [7]. They attributed “quality of design decisions” to “quality of design solution analysis,” which was most directly affected by the availability or unavailability of information [7].

> If the external conditions are characterized by time pressure, the resulting subjective time pressure of the individuals has additional negative effects on information transfer, analysis, and decision making activities [7].

This last finding from the study supports the following conclusion: Designers will rely upon whatever information they may have available in order to construct a design decision when it is clear that time constraints have made reaching an immediate decision imperative.

To conclude this dialogue, it should be noted that the underlying motive is not to portend that design engineers are lazy information gatherers or reckless decision makers. The truth is quite the opposite. Otherwise, design engineers would not expend such vast portions of their overall time and effort seeking and producing design information, and such comparatively little time actually designing. The "default response" of falling back on prior experience or conjecture is elicited by many factors such as schedule constraints, pressure from other colleagues, and task prioritization within the design process. The purpose behind this discussion has been to reveal how instrumental the availability of information is to the design process; and to understand the interactive contexts in which it is most valuable so that information can be managed with methods that will maximize the value of collaboration rather than transforming it into an impediment or burden.

3.2. INFORMATION MANAGEMENT FOR THE HETDEX PROJECT

The need for collaboration and information sharing, and amongst whom, within the HETDEX project was established in section two of this report. This section has discussed the importance of information to the collaborative design process and how the cooperative context determines the real value of information and recommended method of transfer. Examples from the HETDEX project will now be introduced to illustrate these ideas.

The Prime Focus Instrument Platform, or PFIP, could be considered the most crucial subsystem included in the HETDEX upgrade by virtue of its interfaces alone.

The PFIP shares its main interfaces with the Tracker and it includes the Wide Field Corrector. The WFC shares mechanical interfaces with the PFIP structure and Tracker hexapod, as well as optical interfaces with the HET primary mirror and scientific instruments (Figure 6). The WFC collects and focuses the light that is reflected from the immense light-gathering surface of the primary mirror, thus it can be thought of as a set of prescription lenses which enable the HET's instruments to view the stars. If any of these system designs differ by even the slightest amount at their interfaces, mechanical or optical, the HET's ability to carry out scientific research could be compromised.

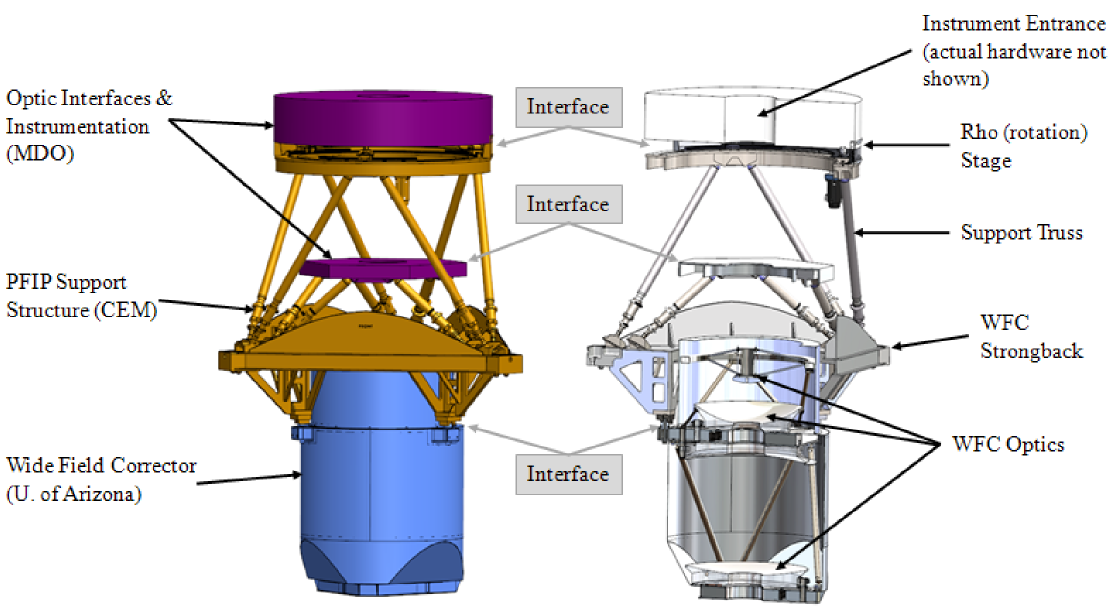

Figure 6: The Prime Focus Instrument Platform (right side view is sectioned at the mid-plane) interfaces and organizational division of responsibilities (CAD content created by CEM and University of Arizona).

The interfaces that reside in the PFIP also signify some of the most critical design collaborations occurring within the project. Engineers from McDonald Observatory, the Center for Electromechanics, University of Arizona, and ADS International must all contribute design information necessary to complete the engineering of this subsystem.

Transactional and collaborative design information transfers between these parties have been critical to design progress.

Transactional information transfers are facilitated through the use of Product Data Management (PDM) software produced by the SolidWorks™ Corporation. The SolidWorks™ Enterprise PDM product is integrated with the SolidWorks™ 3D Computer Aided Design (CAD) modeling environment. PDM enables networked users to upload model and design data, documentation, and virtually any format of support file e.g. presentation, spreadsheet, photos, etc. to a central fileserver (referred to as the "Vault"); while also providing ownership information, user access rights, file revision control, and a number of data reporting features (see Figure 7 below).

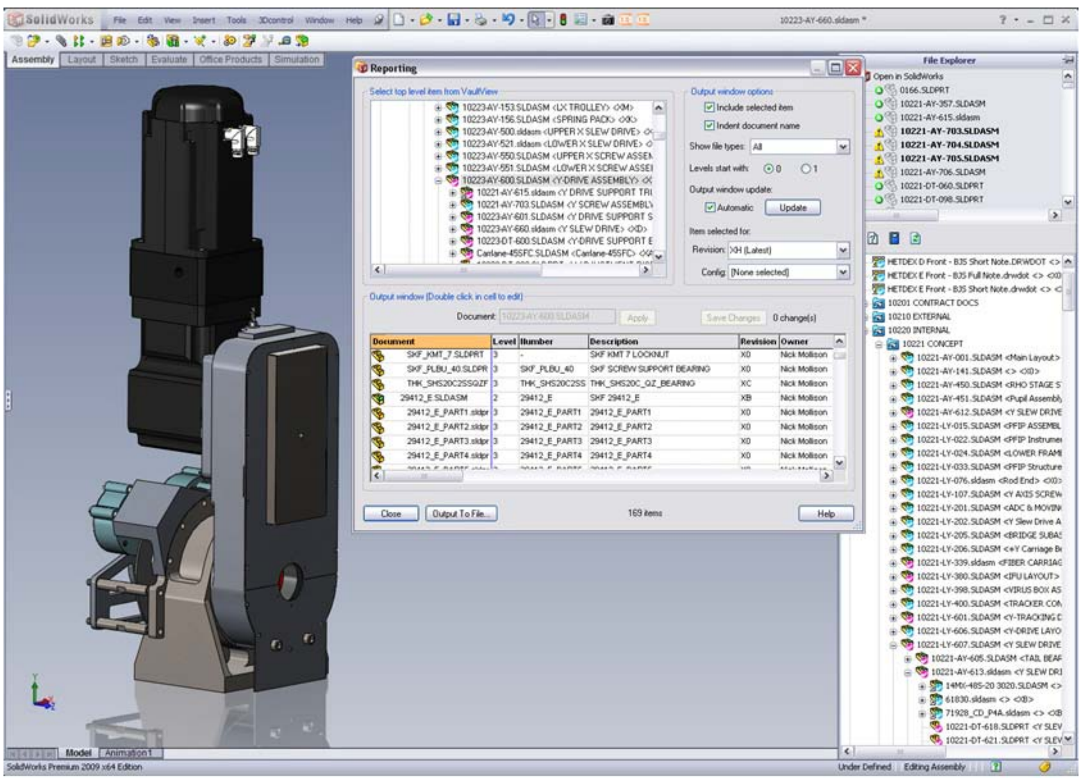

Figure 7: The SolidWorks™ PDM integrated user interface permits document reporting and tracking, and enables users to update component model information in real-time and from within the modeling environment.

The use of PDM and similar tools has become widespread and nearly essential to engineers and designers in sectors such as mechanical, software, and electronic product development [14].

The HETDEX Vault is administrated by CEM. Early in the design phase, a crucial decision was made for McDonald Observatory engineers working on the HETDEX project to adopt the same design software platform and share Vault access rights with CEM[5]. This has enabled direct engineer-to-engineer design information transfers between these two organizations. Outside organizations like the University of Arizona and ADS Intl. are indirectly linked through a slightly less efficient, but still highly effective transfer process whereby technical data packages are sent electronically to a designated point of contact within CEM, and then transferred to the Vault for use by team members. Most often this is accomplished through secure internet FTP transfer, where an FTP host address is created to serve as an electronic “drop-box” for the duration of the project.

Integrating PDM with CAD has enormous benefits for parallel and collaborative design. The process of uploading CAD model data to the Vault, referred to as “vaulting,” is a many-times-daily occurrence on the HETDEX project. The most practical implication is that any time a design team member updates a component design in the Vault, another user working in a CAD assembly containing that component model can, in real-time, execute the “reload” command and the component will automatically update on-screen with virtually no other interruption to productivity. PDM is most appropriate and most effective for this type of passive information exchange [14] or, as Schrage makes the distinction, to transfer information rather than knowledge [17]. Although notes and supporting documents can be attached to CAD information, the primary function of PDM is to enable the transfer of technically rich data in a controlled and

[5] The decision to adopt a common CAD and PDM software platform has had additional benefits that extend beyond data management. Although most commercial CAD software platforms have the ability to convert model data saved in their native file format to a standardized file format (such as STEP or IGES), the accuracy of converted models is not guaranteed, and the ability to exchange conveniently editable models is often lost thereby reducing their utility for collaborative work.

sequential manner [14]. It is transactional in the sense that a discrete parcel of data is packaged, transmitted, and received by one who may then process the information "as-received." Otherwise, supplemental communication such as telephone or email must be initiated in a secondary exchange. Therefore, PDM is a highly efficient tool for managing technical interchanges, but in the collaborative environment it has limitations for managing conceptually rich information and enabling participatory exchanges.

Data from the research study introduced in the section of the report preceding this one suggested that nearly 90% of "critical" design events and decisions occurred in participatory design collaborations. The next example from HETDEX recounts an event which demonstrates this conclusion. Section two of this report made reference to the number of stakeholders and considerations that are represented in the set of Tracker design objectives. One pivotal event in the Tracker design process contributed more design and interface decisions in advancement toward overall design objectives than perhaps any other. This event was a Failure Modes and Effects Analysis (FMEA) conducted jointly between CEM and MDO. This face-to-face meeting initialized with both teams near-fully represented in the same conference room. The few who could not be present were able to participate via telephone and internet conferencing. The FMEA process involves agreeing upon rating scales for failure likelihood within a given time period and resulting magnitude of injury in terms of dollars, assets, or to personnel. Next, the major failure modes of each subsystem are identified, and the effects of failure to the overall system are determined. In accordance with the severity of outcome and likelihood, the rating scales are applied and a final risk rating is given to each failure mode. After the ratings have been compiled, a remediation plan for the design is constructed and priority is assigned (Table 2).

Table 2: An example taken from the HETDEX Failure Modes and Effects Analysis exercise. Note: "RPN" refers to Risk Priority Number, which is used to prioritize risk mitigation design tasks.

Item	Potential Failure Mode	Potential Effects of Failure	Severity	Potential Cause/ Mechanisms of Failure	Occurrence	Current Design Controls Prevent/Detect	Detection	RPN	Recommended Actions
2.1	Tracker bridge exceeds all travel limits and decouples from upper hexagon structure.	Damage to tracker bridge, telescope structure, mirrors and possibly to staff.	**5**						
2.1.1				Accidental power up of drive motors	3	- Drive motors can only be powered up if both Tracker Computer (TC) and Telescope Control System (TCS) agree (presumed) - TCS detects out of bounds following error and commands shutdown (would not apply if this is not tracking event) - Software limits prevent tracker from exceeding travel limits	1	15	Implement torque limiting device in drive system. TBD as to whether this is electrical or mechanical device.

Conducting an FMEA with the level of detail and number of parties represented by HETDEX is obviously time intensive. The full-group meeting took place over two days before splintering into assignments between pairs consisting of one CEM and one MDO engineer. As the results of the collaborative design research study would imply, the outcome of this collaboration was a number of highly important adjustments to the Tracker design. Some were minor in scope, such as revisions to safety switches and control interlocks, while others redefined interfaces and added or removed whole subsystems. Another significant outcome was an overall refinement to the functional requirement definition of several subsystems which still contained a degree of ambiguity beforehand. These types of design decision outcomes stand in stark contrast to those that would, or even could be enabled with passive information transfer. The heart of the FMEA was clear elicitation and explanation of design requirements, which could only take place interactively, and proportionate weighting reflects their real value to the stakeholder.

3.3. Information Management Recommendations

One of the conclusions reached in the HETDEX project example is that PDM is a highly efficient tool for managing technical interchanges, but in the collaborative environment it has limitations for managing conceptually rich information and enabling participatory exchanges. The effect this last statement has on the design process, in the context of a PDM-linked workgroup, needs to be understood because system complexity will only make it more crucial. This environment, coupled with a flat organizational structure (common in many engineering and design project organizations), means passive information transfer will frequently occur at the lowest nodes of the network i.e. at the level of the individual responsible for a particular subsystem or component. This has the implication that visibility into the social domain of the system described earlier may be limited at the team and manager level. Changes manifest in the components of the system. Each designer becomes aware of the changes as components are updated in iterative and sequential fashion. They make the newly required modifications to their component and send it up to the network. If things were left this way, without supplemental communication, change propagation two or more persons removed from the change originator would be virtually invisible and therefore rarely communicated upward to the manager or to the team at-large.

The effect "good solution analysis" and information availability have on design quality and cost was another significant conclusion from collaborative design research [7], [15]. PDM can significantly benefit design quality by virtue of enabling common and consistent design information so that designs will satisfy their requirements at their interfaces. Cost is a separate issue. In terms of information transfer efficiency and quality after the design is transferred to manufacture, PDM is a benefit. However, it can also contribute significantly to project cost by obscuring excessive design iteration and diverting attention away from interface and solution analysis by fixating individual designer's attention on component design issues.

Participatory collaboration can be time consuming and carries with it a substantial project burden when full team member presence is required. However, it is also instrumental to early and late "critical situations" within the design process. When such collaboration environments are scheduled to materialize ahead of time, such as design reviews and team meetings, their value should be maximized by reporting and probing architecture level design decisions to gain greater transparency into the dynamics of the social system domain. They are also the most effective environment to initiate participatory exchanges to resolve uncertain or ambiguous information that may be impeding design progress.

These conclusions create the framework for the central management recommendation. Information needs and medium of communication should be considered early in the design project planning phase, and throughout the project as it develops. This seems as though it would be obvious, but how often do "information needs" really take center stage before the subjects of resource levels, task durations, and budgets in initial project planning? It has been shown that seeking, accessing, and processing information is a significant factor in total process time, rework iteration, and quality, and will therefore determine a large share of the design project's schedule and capital needs. The decision to use a common CAD and PDM platform between collaborators on the HETDEX project was more than just a sensible choice. There were costs for the other party to adopt the software in the form of license fees and time required to adapt, but the costs are heavily outweighed by the rapidity and quality of design information exchange that it has enabled. Danilovic and Sandkull summarize the questions that must be asked:

> Who needs to communicate to whom in order to solve their tasks? What kind of information needs to be exchanged? Why is this information exchange important to other people? When should this information exchange take place? How

should people involved share the needed information with each other in order to handle interdependencies? [4].

The question of how should information exchanges take place refers to the choice of medium and process, which requires the answer to at least two additional questions: Is the information needed in order to resolve substantial uncertainty or ambiguity? Is the real need one of transferring the required information, or actually creating the required information? When uncertainty is high and/or the objective is to create information, interactive and participatory exchanges should be initiated, as illustrated by the HETDEX FMEA example. When the inverse is true, passive and automated communication methods are suitable and likely to be more efficient. A final factor to consider is the frequency with which the information will be accessed and by whom. With high frequency and number of designers requiring access, the cost of documenting and consolidating information is often far less than the cost of repetitive searches, though this is not always true. For example, attaching supplier information, purchase dates, and manufacturing cost information to every component in a PDM database is a valid effort for recurring engineering and design projects, but it would not necessarily provide positive earned value to a one-time project or to a collaborative team that will dissolve upon project completion.

In light of the recommendation to bring information exchange and communication needs to the forefront of collaborative project planning, a practical means of identifying those needs at an early project stage will be required. One such method will be introduced in subsequent sections.

4. Complexity Management

If higher degrees of system and project complexity make collaborative design more difficult to manage, it would seem that the obvious solution would be to develop methods to simplify the systems involved. This is a laudable goal, and one that may even be practical to accomplish in some scenarios. The word “complexity” has been used many times throughout this report, and further understanding of the underlying concept within the context of collaborative engineering and design of complex systems will expose the inherent difficulty of “simplification.” Complexity as presented in this report means, at its core, that prodigious changes to the order and structure of any system which possesses sufficient quantities of elements and strength of interrelationships to actually be deemed “complex” will only create new interrelationships that are no less easy to manage or more immediately transparent in their implications. Put another way, it may be possible to reduce or isolate complexity at local nodes within the system, but drastic simplification of interfaces and relationships would create impractical requirements for the component and its designer to fulfill, or manifests in an undesired modification to the global function of the system ([12], [16], and [18]). The resulting conflict is often encountered in complex systems that require high reliability. Simplification is necessary to reduce the number of functions one subsystem is required to perform, therefore reducing that subsystem’s probability of failing to perform at least one of its functions [18]. If functional simplification cannot be achieved without altering the function of the global system, then it may be necessary to have redundant systems performing the same set of functions [18].

The above conflict arises in tightly integrated systems, and especially those that involve functional overlap i.e. more than one subsystem contributes to a single system-level function [18]. The same qualities contribute uncertainty – the next element of difficulty encountered in complex systems. Technical and social uncertainty both grow with the number of elements in the system due to the number of system variables and respective strength of influence over system qualities [4]. “The variability of the system will approach the variability of its most variable component” (Reinertsen, [16]). A structurally stable technical and social system will attenuate the disturbances caused by small design changes, and an unstable one will amplify them. Section two of this report showed how system architecture contributes to complexity. Section three illustrated that the complexity of the architecture drives organizational needs for information and communication, which are immensely critical to collaborative design. This section of the report will show how the very technical explanation of complexity given above translates to the practical world of design management. Namely, how complex systems can be visualized, understood, and manipulated even when they are not readily simplified.

4.1. DEPENDENCY: THE SOURCE OF DIFFICULTY IN COMPLEX PROBLEMS

Restating the central issue: complexity is a product of the quantity of interfaces and the relative magnitude of interdependency each interface creates between components. Returning to the PC design example will help illustrate this point. In the example, the supply voltage was listed as one parameter shared at the interface between the video adapter card and the motherboard; i.e. the voltage received by the video card depends on how many volts the motherboard supplies to the connector. Suppose that the motherboard design is in its preliminary phase and the motherboard designer knows only that it will supply some specific value within a range of 1 to 12 volts. In this scenario, the video card can be completely configured with components that will all function normally with a supply voltage anywhere between 1 and 12 volts. The fact that the motherboard designer can only provide highly uncertain information at this time is almost immaterial to the video card designer. A refined estimate of the exact supply voltage would contribute little value because the video card’s design is not dependent upon which value between one and twelve is ultimately decided. Stated formally, resolving uncertainty at the information source is less important when minimal dependency exists at the information receiver. Now consider the same scenario with the exception that the video card designer must choose only one of two possible sets of subcomponents. The first of these sets will only operate on 1-6 volts, and the second on 6-12 volts. Now the video card's design is highly dependent upon the estimated supply voltage and it's unlikely that components can be specified without first obtaining better information. Resolving uncertainty at the information source is critical to the information receiver when stronger dependency exists.

The interplay between variability and dependency is the tool with which complex systems can be manipulated to manage collaborative design productivity. In the PC design example from section two of this report, the interface parameter of the video card connector's orientation on the motherboard was used to illustrate the difference between the two-component scenario and the broader system scenario, and it shall do the same here. The dependency between the video card's orientation inside the PC and the orientation of the connector at its interface was a full one-to-one correlation. Rotating the connector some number of degrees rotates the video card by the exact same amount. Furthermore, other dependencies were found to exist between other components at this interface once the design change was implemented. The variability brought to light by the motherboard designer's change had maximum impact at the system level (in both the technical and social domains) because the dependency correlation was very strong and extended across multiple interfaces. If the variability and complexity surrounding the connector interface had been better understood beforehand, it might have been adjusted accordingly. One possible solution would be to reduce the strength of dependency correlation at that particular interface. The solution might be as simple as linking the video card to the motherboard with a flexible cable instead of fixed connectors, so that it can more easily satisfy numerous other system constraints without successive rounds of design iteration.

Proposing that interfaces should be evaluated and adjusted in terms of intangible attributes like variability and dependency relationships is a high-level and perhaps even esoteric way of thinking that seems to overlook the obvious power of design intuition and ingenuity, so what added value does it generate? The value lies in its ability to provide traction in addressing the problem with extremely complex systems – which is that the solutions are no longer intuitive. Or more accurately, the full downstream consequences of a proposed solution are much more opaque and difficult to evaluate than it is to recognize when an intuitively derived solution seems to address the immediately visible design problem. Yet, it is easier and more effective over the full term of the design process to start with the right questions than it is to construct more and more solutions as each one reveals new and unforeseen systemic faults [4].

> The importance of interfaces is often overlooked because we are constantly lured away by the concrete nature of module design ...When you have a design problem, and ask what needs to be changed to fix it, the answer is always a component, so you conclude that the component was the cause of the problem. In reality, for many systems the root cause of component redesign is really poor interface design (Reinertsen, [16]).

The compartmentalization of responsibility inherent to collaborative work makes this an even easier pitfall to slip into, and there are often reasons why it may be impossible to avoid on a given project. The partitioning line between subsystems becomes an increasingly concretized divider of group and organizational responsibilities as the project progresses. Therefore, significant external pressure exists for issues developed or manifested within a subsystem to be resolved by the responsible owner, and with relatively little attention paid to the interface which is perceived as being rigid [16]. As alluded to, there are some instances where the interfaces must be inflexible, such as those whose constraints extend to the social domain where the interface also demarcates a separation of domain knowledge or contractual obligations, for example. It is very likely that managers and designers involved in collaborative design projects will be forced to live with at least some rigid interfaces and their attendant dependencies. Therefore, the need to understand and communicate the implications of such dependencies to the opposite stakeholder, and also to manipulate system variables and structure within one's own boundaries, becomes highly important to mitigating negative dependency constraints and maintaining productivity.

4.2. HETDEX Complexity and Tools for Management

Understanding and managing complexity is one of the most difficult challenges posed by the collaboration to design a new Tracker for the HETDEX upgrade. The Hobby Eberly Telescope is, itself, a very sophisticated machine that relies upon numerous mechanical, electrical, and computerized subsystems to carry out the complex and precisely orchestrated function of tracking and measuring starlight emanating from incredible distances away. There are also numerous design factors that influence how well the telescope will be able to perform this function which must receive consideration; from the speed and precision of its servo-drive mechanisms, down to what type and brand of paint is used on them (in order to prevent unwanted light reflection). The nature of the project, which is to enable new scientific discoveries, also pushes the design process to achieve some technical objectives that approach the extreme outer boundary of what's possible under current engineering research and development projects. Many new and novel solutions to new design problems are required and system-level integration and optimization present significant challenges. Because the Tracker system architecture is complex and highly interlaced in terms of both interfaces and design requirements, design decisions require hefty consideration and oftentimes the involvement of several team members to represent each relevant domain knowledge area. Section three provided examples of how different types of information and information exchanges were utilized in the HETDEX project. The discussion that follows in this section will introduce a method for evaluating the project's technical and social information in order to analyze design decisions, and to understand their implications in the context of the system design and the collaboration environment.

Despite the fact that many of the Tracker's major subsystems had been established by its predecessor designs, the redesign of the Tracker system for HETDEX has led to revisions in the quantity and location of subsystem interfaces, and even the number and type of subsystems present on the Tracker. Efforts were made to focus on these types of decisions and restrict them as much as possible to early design phases, but evolving system designs in research and development project environments will inevitably lead to some changes occurring much later than would be ideal. Of these, some may precipitate heavy rework penalties, while others may be cost-free or even cost-saving. The number and strength of dependencies affected within the system determine which category will describe a proposed design change [12]. When design involves collaboration and concerns a complex system such as the Tracker, the shortcomings of intuitively evaluating design decisions and their consequences, which were described above, become apparent. Intuitive design evaluation methods are prone to overlook less obvious system interactions, meaning awareness of their consequences will not exist until after the change has been implemented. Included in this set of consequences are the interactions in the social domain, which are very difficult to perceive intuitively. A primary motivator in researching the topic of complexity management for this report was to uncover practical methods for understanding the complexity of the design environment and systematically evaluating design change implications over a more complete spectrum. Unlike the use of PDM for information management, the following complexity management tool has not been implemented on the HETDEX project prior to the work performed for this report and development of its application is ongoing.

The Dependency Structure Matrix (DSM), also referred to as a design precedence matrix, problem solving matrix, and design structure matrix to name a few of its aliases [11], is a graphically represented model of system dependency interactions. Constructing the basic matrix is fairly straightforward, but its real contribution of enabling various forms of system analysis is performed using more complex mathematical algorithms. A detailed discussion regarding the construction and use of dependency structure matrices is outside the scope of this report, though additional information can be found within the appendix and references [4] [5] [11] [12] and [13]. A DSM for a selected sample of the Tracker's major component subgroups is shown in Figure 8 below.

Name	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Lower X-axis bearing trolley	1	1															
Lower X-axis screw drive	2	X	2														
Upper X-axis bearing trolley	3			3													
Upper X-axis screw drive	4			X	4												
Tracker bridge	5	X		X		5	X			X							
Y-axis screw drive	6					X	6		X								
Y-axis bearing trolley (Left)	7							7									
Y-axis bearing trolley (Right)	8								8								
Lower hexapod frame	9							X	X	9	X						
Hexapod struts	10									X	10	X					
Upper Hexapod frame	11											11			X		
Wide Field Corrector (WFC)	12									X	X		12	X			
WFC strongback	13											X		13			
PFIP support structure	14														14		
PFIP upper hexapod	15															15	X
PFIP Rho stage	16														X	X	16

Figure 8: Each 'X' in the Dependency Structure Matrix for the baseline HETDEX Tracker design reflects mechanical interfaces shared between major component subgroups corresponding to each number, as well as directionality of design change propagation. Cells shaded red indicate co-dependencies and subsequent potential for iterative design rework.

One obvious application of DSM is to provide qualitative comparison of the dependency structures between various design alternatives. Figure 9 reflects a new dependency structure after several design changes were made intuitively during the design's evolution. To summarizing these changes: The subgroup 'PFIP Upper Hexapod' was eliminated from the design. 'Upper Hexapod Frame' and 'WFC Strongback' were merged into one subsystem. Finally, the 'Lower Hexapod Plate' component was added and provides a buffer interface between 'Lower Hexapod Frame' and 'Hexapod Struts.'

Name	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Lower X-axis bearing trolley	1	1														
Lower X-axis screw drive	2	X	2													
Upper X-axis bearing trolley	3			3												
Upper X-axis screw drive	4			X	4											
Tracker bridge	5	X		X		5	X			X						
Y-axis screw drive	6					X	6									
Y-axis bearing trolley (Left)	7							7								
Y-axis bearing trolley (Right)	8								8							
Lower hexapod frame	9							X	X	9						
Lower hexapod plate	10									X	10					
Hexapod struts	11									X	X	11	X			
Upper hexapod frame / WFC Strongback	12												12		X	
Wide Field Corrector (WFC)	13									X		X	X	13	X	
PFIP support structure	14														14	
PFIP Rho stage	15														X	15

Figure 9: HETDEX Tracker Dependency Structure Matrix following changes to the component subgroup architecture.

The second Tracker DSM reveals the modified architecture contains fewer co-dependencies. However, the total number of dependencies is only reduced by two. Conclusions as to why this is true can be drawn from the two figures: Completely eliminating one subgroup eliminated all of its associated dependencies; merging two subgroups redistributed dependencies; and introduction of a new subgroup, called ‘Hexapod Plate,’ between two interfaces created new dependencies. Insight as to why ‘Hexapod Plate’ was added will underscore the value of DSM methods. ‘Lower Hexapod Frame’ was a complex and multi-functional component under the stewardship of CEM in Austin, Texas. The ‘Hexapod Struts’ subgroup design was engaged by ADS International of Italy, and execution of its design began at a later date. According to Reinertsen, “Communications will be worst if we compound the problem by placing an undefined interface on top of both an organization boundary and a geographic boundary” [16]. Due to the later start date and communication boundaries, this is a scenario where variability is high and means to resolve uncertainty are restricted. The ‘Hexapod Plate’ component subgroup added structural complexity by adding interfaces, but doing so also meant the co-dependency between the two interfaces was resolved, and the variability was then confined to one simple and easy to change component. Design iteration and each designer's dependency on timely information transfer in order to maintain productivity were subsequently reduced through this manipulation of system dependency and variability.

DSM gains additional value when combined with analytical algorithms and software. An example of further DSM capabilities can be found in the Appendix of this report. The DSM examples constructed here were created using basic, freely-distributable software tools. Commercial software has been investigated for future project applications as part of this research. The matrices presented above are "binary," meaning dependencies either do, or do not exist. More sophisticated programs allow for numerical values, permitting variation in the strength of dependency which adds another parameter of system control. The most developed commercial DSM products also support Multi-Domain-Matrices (MDM) which link dimensions of the product, process planning, and social domains and contain algorithms for generating alternative system structures.

Research interest in DSM analysis methods grew sharply in the 1990's, and notable contributions have since been made in complex design environments such as the auto industry ([4], [11], and [12]). The potential benefit of these tools, and likewise the motivation behind their ongoing development, can be seen from the HETDEX example. When system complexity and design dependency can be visualized in a lucid format, systemic consequences of design changes and alternative proposals become easier to evaluate a priori.

4.3. COMPLEXITY MANAGEMENT RECOMMENDATIONS

The concept of complexity was further developed at the beginning of this discussion into the idea that when the number of interrelated system subcomponents is very high, and the interrelationships form an overlapping network rather than a strict hierarchy, the full implications of design choices become more difficult to predict.

Complexity is also shared between the design of components in the technical domain and the collaborative interaction between designers in the social domain. Finally, the root cause of difficulty in managing complex system designs lies in interfaces and interrelationships that create dependency, either with respect to design information or the embodiment of the design solution.

Recommendations for complexity management begin first with a comment on how we view design, especially in parallel and collaborative design situations. Traditional project management tools, such as Gantt charts, are able to incorporate aspects of design task dependency and overlap, yet they still represent a very linearized view of the design process. The Project Evaluation and Review Technique (PERT) improves upon this view by transforming similar project management information into a "node and vector" network of tasks, which is a closer representation of the collaborative network. These two methods concentrate on critical path management, however, where the objective is to create forecasts based on resource usage and task durations. These two variables are manipulated to satisfy an estimated project completion schedule and budget. When information is updated, the changes propagate serially from the origin of change to the end of the project (when the change has influence over the critical path), hence the term Critical "Chain" Project Management (CCPM), or Critical Path Management ([4], [13], and [14]). The shortcoming of such CCPM methods exposed by highly complex and collaborative projects is this "chain" representation where events feed-forward through the chain. Although task dependency can be represented relatively well with Gantt and PERT models, they provide precious little insight for managing information flow and iteration due to the absence of task feedback and looping ([4], [13], and [14]). It has been shown that additional complexity in technical and social systems introduces more subtle dynamic behavior. Changes "ripple and bounce" rather than "flow," by virtue of generating both feed-forward and feedback responses throughout the system.

The first recommendation is a generalized conclusion from the discussion above: Complex collaborative design projects must be thought of as being different from design projects with simpler interactions and much less uncertainty. Obviously, this wouldn't be a very useful recommendation without suggesting in which ways the two differ. Returning to the statement: “The variability of the system will approach the variability of its most variable component” [16]; Reinertsen’s assertion is based on the principle that a sufficiently large collection of variable designs or design tasks will converge on the variability of the least certain element, because the number of positive outcomes and number of negative outcomes will, for the most part, average over the course of the design process [16]. It is for this reason that CCPM methods are still able to provide validity when the tasks are highly variable, but sufficient in quantity to offset each other. However, complex collaborative projects require an alternative, or at least supplemental viewpoint because: 1) Variability is an input to, not an output from CCPM methods and therefore it does not generate any additional insight into managing the variable element itself, and 2) The “law of averages” outlook inherently dilutes information such that it provides much less utility and accuracy in evaluating specific design decisions at discrete points along the way. Furthermore, it does not fully account for variability or dependency that fluctuates and migrates as the design process unfolds, which occurs regularly in the R&D setting; and there is little to bring “self-energized” variables and rework loops hidden within the system architecture to light.

Additional planning tools and visualization methods, such as the Dependency Structure Matrix methods discussed in the previous section, should be employed in such scenarios where complexity and variability are high. Multi-Domain-Matrix tools go a step further and provide even finer granularity for evaluating design decisions in the technical domain, social domain, and design process planning domain. The final, and most important point of difference behind this recommendation, is that dependencies must also be brought to the forefront of collaborative design management in tandem with information management. Information needs are both an input to, and output from the dependency matrix. Identifying information dependencies which exist as a result of structural complexity aids information exchange process planning, and provides the necessary picture of rework potential and rework impact that will augment traditional CCPM planning in collaborative efforts.

It is vital that the nature of system dependencies be understood, and that they can vary both in strength and directionality. The ability to evaluate system relationships in a quantitative and systematically rigorous way with DSM positively supplements intuitive design evaluation with thoroughness, objective and consistent comparison of alternatives, and utility in formulating pro-active design architecture decisions.

The second set of recommendations is a more practical application of complexity theory, and already familiar to many in design practice. One of which has already been practiced on the HETDEX project in the form of incorporating uncertainty margins for individual component mass budgets. The uncertainty margins were then reduced as designs became more mature. It was stated before that dependency and variability were two parameters that could be manipulated to manage complexity. Design margins are one such method of reducing dependency correlation across interfaces, and it therefore creates a buffer to absorb a given amount of variability [1]. The corollary to design margins is: When dependency cannot be adequately reduced and relative uncertainty is very high, it may be possible to isolate them from other components in the system through appropriate interface selection ([1], [12], [16] and [18]). The PC design example achieved this by adding the ribbon cable component, which decoupled two component interfaces, as did the addition of the 'Hexapod Plate' component in the HETDEX project. A third option is to exercise managerial options with respect to design task prioritization. It is possible to satisfy many design objectives and dependencies with partially complete designs. There is typically no requirement that a design's evolution must be continuous, although there are some negative consequences associated with interrupting design progress. This option becomes attractive in the aforementioned instances when externally imposed interfaces are rigid and uncertainty is high. The aim is to avoid "overinvesting" in a design such that only the necessary design information is produced and reflexive variability can impose no more than the minimum amount of rework. All of these recommendations are practical incarnations of Suh's logical theorem to reduce "system stiffness" [18] i.e. it's resistance and sensitivity to change.

In summary, complexity management should concentrate on understanding the dynamics of the collaborative process, making design solution decisions in accordance with a system-based perspective, and remaining flexible and adaptable when feedback begins to flow in. Tools such as DSM and MDM analysis exist, and should be used to pinpoint critical decisions and collaborative exchanges to be monitored by management, as well as supplementing intuitive decision analysis.

5. Conclusion

Schrage asserts that we engage in collaboration when faced with a problem that we're unable to solve on our own [17]. This is the real value of collaborating in R&D design projects – To engage more difficult problems and elevate ingenuity beyond the capabilities of any single group or organization involved. There are many buzzwords presently circulating the design management community collectively referred to as "Design for X," where X may be quality, manufacture, reliability, etc. The concept of viewing collaborative design as a technical system and social system discussed in this report implies that we should also "design for collaboration." Just as designers are responsible for the design of systems and components, managers are responsible for designing the framework of interactions, processes, and monitors that transpire in the social domain.

The important conclusions contained in this report are as follows: The selection of interfaces between components, subsystems, people, and organizations which comprise the system architecture will determine the information and communication needs of the project. Information gathering and processing is perhaps the most critical factor determining design productivity and quality. Complexity makes gathering the right information at the right time more difficult, and obscures the impact such information will have once it is received. The challenges of managing collaborative design revolve around uncertainty and potential for hindered or wasted effort as the number of variable elements and people involved grow in number. Information challenges can be addressed by recognizing what needs to be exchanged, how the exchange should occur, and when, based on the dependency relationship and whether the need is for transfer or creation. Complexity challenges can be addressed similarly, by recognizing dependency relationships at an early stage, determining whether or not they are directional or iterative, and prioritizing to either resolve uncertainty or mitigate the strength of dependency so that designers may continue in their efforts unimpeded.

Appendix

This appendix is intended to provide further explanation and expand upon the Dependency Structure Matrix example introduced in section 4.2 of this report. For comprehensive discussion of dependency matrix methods consult references [4] [5] [11] [12] and [13].

The baseline HETDEX Tracker DSM was constructed by analyzing the flow of mechanical design information between components and subsystems. Figure A1 illustrates how directionality of information is determined from the DSM.

	Name	Lower X-axis bearing trolley	Lower X-axis screw drive	Upper X-axis bearing trolley	Upper X-axis screw drive	Tracker bridge	Y-axis screw drive	Y-axis bearing trolley (Left)	Y-axis bearing trolley (Right)	Lower hexapod frame	Hexapod struts	Upper Hexapod frame	Wide Field Corrector (WFC)	WFC strongback	PFIP support structure	PFIP upper hexapod	PFIP Rho stage
Name	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Lower X-axis bearing trolley	1	1															
Lower X-axis screw drive	2	X	2														
Upper X-axis bearing trolley	3			3													
Upper X-axis screw drive	4			X	4												
Tracker bridge	5	X		X		5	X			X							
Y-axis screw drive	6					X	6		X								
Y-axis bearing trolley (Left)	7							7									
Y-axis bearing trolley (Right)	8								8								
Lower hexapod frame	9							X	X	9	X						
Hexapod struts	10									X	10	X					
Upper Hexapod frame	11											11			X		
Wide Field Corrector (WFC)	12									X	X		12	X			
WFC strongback	13											X		13			
PFIP support structure	14														14		
PFIP upper hexapod	15															15	X
PFIP Rho stage	16														X	X	16

Figure A1: HETDEX Tracker baseline Dependency Structure Matrix depicting direction of information flow.

The arrows represent the logic that should be followed to read the DSM. When beginning with the row, the item in the row is said to pass information to the item in the column when the two share a dependency indicated by a mark in the cell shared by the row and column. Marks above the diagonal feed information forward, and marks below the diagonal feed information backward, as indicated by the sequence in which they appear in the matrix. The DSM in Figure A1 would be interpreted as: The Lower Hexapod Frame receives information from the Tracker Bridge, and sends information to the Y-Axis Bearing Trolley. If the sequence of the matrix is considered, the relationship can be further decomposed as follows: The Tracker Bridge feeds information forward to the Lower Hexapod Frame. The Lower Hexapod Frame feeds information back to the Y-Axis Bearing Trolley. Figure A2 provides a block diagram of this relationship.

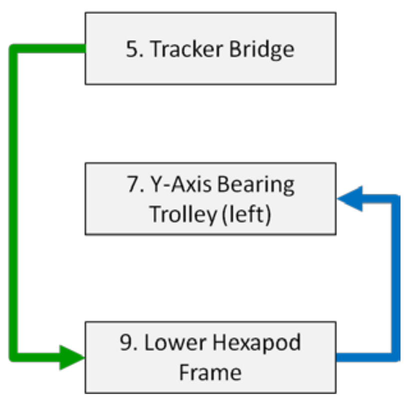

Figure A2: The information flow of the Tracker DSM in Figure A1 represented as a block diagram. Information feed-forward and feedback would occur if design activities were sequenced this way.

It can be seen from the block diagram representation that the DSM may be used to facilitate identification of iterative information dependencies and potentially less than ideal task sequences depending on the direction of information flow. The term "partitioning" the matrix means reordering the entries such that feed-forward relationships are maximized and feedback dependencies are minimized (Figure 3). This can be accomplished manually, but may be impractical for large matrices and therefore the use of software is preferred. A freely-distributable spreadsheet software add-in (available from [11]) was used to construct the examples that follow.

New Seq.	Name	#	1	3	7	8	14	2	4	11	15	16	9	10	13	5	6	12
16	Lower X-axis bearing trolley	1	1															
15	Upper X-axis bearing trolley	3		3														
14	Y-axis bearing trolley (Left)	7			7													
13	Y-axis bearing trolley (Right)	8				8												
12	PFIP support structure	14					14											
11	Lower X-axis screw drive	2	X					2										
10	Upper X-axis screw drive	4		X					4									
9	Upper Hexapod frame	11					X			11								
8	PFIP upper hexapod	15									15	X						
7	PFIP Rho stage	16					X				X	16						
6	Lower hexapod frame	9			X	X							9	X				
5	Hexapod struts	10								X			X	10				
4	WFC strongback	13								X					13			
3	Tracker bridge	5	X	X									X			5	X	
2	Y-axis screw drive	6				X										X	6	
1	Wide Field Corrector (WFC)	12											X	X	X			12

Figure A3: The Tracker DSM after it has been partitioned (reordered) to maximize forward information flow. The new sequence is indicated at the left, and proceeds from the bottom up.

The sequence of the original matrix was chosen somewhat arbitrarily based upon the spatial location and grouping of components on the Tracker, not by the order of design tasks. DSM may also be used in this manner to derive a baseline task sequence, though many additional factors may need consideration and pairing with Gantt or PERT methods may provide additional value. Note that the software routine used here employs a “lower triangular” convention and moves dependencies below the diagonal, therefore the sequence is ordered from bottom to top. An “upper triangular” convention is also frequently used [11], and each may find a preference depending on the application e.g. software design versus mechanical design.

"Banding" a DSM (Figure A4) refers to identifying groups of elements that are independent from each other. The items within each "band" do not share any information dependencies with each other, and if the items represent tasks they may therefore be conducted in parallel with each other.

Name	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Lower X-axis bearing trolley	1	1															
Lower X-axis screw drive	2	X	2														
Upper X-axis bearing trolley	3			3													
Upper X-axis screw drive	4			X	4												
Tracker bridge	5	X		X		5	X			X							
Y-axis screw drive	6					X	6		X								
Y-axis bearing trolley (Left)	7							7									
Y-axis bearing trolley (Right)	8								8								
Lower hexapod frame	9							X	X	9	X						
Hexapod struts	10									X	10	X					
Upper Hexapod frame	11											11			X		
Wide Field Corrector (WFC)	12									X	X		12	X			
WFC strongback	13											X		13			
PFIP support structure	14														14		
PFIP upper hexapod	15															15	X
PFIP Rho stage	16														X	X	16

Name	#	1	3	7	8	14	2	4	11	15	16	9	10	13	5	6	12
Lower X-axis bearing trolley	1	1															
Upper X-axis bearing trolley	3		3														
Y-axis bearing trolley (Left)	7			7													
Y-axis bearing trolley (Right)	8				8												
PFIP support structure	14					14											
Lower X-axis screw drive	2	X					2										
Upper X-axis screw drive	4		X					4									
Upper Hexapod frame	11					X			11								
PFIP upper hexapod	15									15	X						
PFIP Rho stage	16					X				X	16						
Lower hexapod frame	9			X	X							9	X				
Hexapod struts	10								X			X	10				
WFC strongback	13								X					13			
Tracker bridge	5	X	X									X			5	X	
Y-axis screw drive	6				X										X	6	
Wide Field Corrector (WFC)	12											X	X	X			12

Figure A4: The un-partitioned DSM (left) and partitioned DSM (right) as they appear after banding. Note that reordering the matrix has enabled more tasks to be potentially conducted in parallel.

Finally, DSM may be coupled with additional analysis capabilities to estimate project cycle-time. Monte-Carlo analysis was performed for the HETDEX example, though the task time estimates were chosen arbitrarily and used only to explore the implications of dependency relationships comparatively. The analysis begins with task duration estimates and a "learning curve" estimate (Table A1). Learning curve refers to the percentage of the original task duration required to complete each subsequent iteration.

Table A1: Minimum, maximum, and most-likely duration estimate for each task, and associated learning curve (percentage of initial duration required to complete every rework iteration thereafter).

Initial Sequence	Activity Name	Duration Min	Duration Likely	Duration Max	Learning Curve (0 to 1)
1	Lower X-axis bearing trolley	10	15	20	0.1
2	Lower X-axis screw drive	20	30	40	0.3
3	Upper X-axis bearing trolley	15	15	20	0.1
4	Upper X-axis screw drive	20	30	40	0.3
5	Tracker bridge	60	65	80	0.3
6	Y-axis screw drive	30	45	60	0.6
7	Y-axis bearing trolley (Left)	10	15	20	0.1
8	Y-axis bearing trolley (Right)	10	15	20	0.1
9	Lower hexapod frame	10	15	20	0.4
10	Hexapod struts	5	10	20	0.1
11	Upper Hexapod frame	10	15	20	0.1
12	Wide Field Corrector (WFC)	70	80	90	0.5
13	WFC Strongback	30	45	60	0.6
14	PFIP support structure	10	20	30	0.1
15	PFIP upper hexapod	10	15	20	0.2
16	PFIP Rho stage	20	30	40	0.4

The next input parameter is the probability of rework (Figure A5). The probability of rework can be made dependent on the direction of information flow, e.g. a change to "component A" may have a higher probability of creating rework in "component B" than vice versa.

Probability of Rework		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Lower X-axis bearing trolley	1	1															
Lower X-axis screw drive	2	0.1	2														
Upper X-axis bearing trolley	3			3													
Upper X-axis screw drive	4			0.1	4												
Tracker bridge	5	0.1		0.1		5	0.3			0.1							
Y-axis screw drive	6					0.4	6		0.5								
Y-axis bearing trolley (Left)	7							7									
Y-axis bearing trolley (Right)	8								8								
Lower hexapod frame	9							0.3	0.3	9	0.8						
Hexapod struts	10									0.7	10	0.7					
Upper Hexapod frame	11											11			0.6		
Wide Field Corrector (WFC)	12									0.4	0.3		12	0.7			
WFC strongback	13											0.7		13			
PFIP support structure	14														14		
PFIP upper hexapod	15															15	0.6
PFIP Rho stage	16														0.7	0.4	16

Figure A5: Percent probability of rework for each component. Direction of information flow is a factor in choosing each value.

The final simulation input is the impact of rework effort (Figure A6). This is the extent of the redesign that would be required should another component with a shared dependency experience changes.

Rework Effort Impact		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Lower X-axis bearing trolley	1	1															
Lower X-axis screw drive	2	0.6	2														
Upper X-axis bearing trolley	3			3													
Upper X-axis screw drive	4			0.6	4												
Tracker bridge	5	0.2		0.2		5	0.5			0.7							
Y-axis screw drive	6					0.3	6		0.4								
Y-axis bearing trolley (Left)	7							7									
Y-axis bearing trolley (Right)	8								8								
Lower hexapod frame	9							0.3	0.3	9	0.4						
Hexapod struts	10									0.3	10	0.3					
Upper Hexapod frame	11											11			0.6		
Wide Field Corrector (WFC)	12									0.6	0.5		12	0.8			
WFC strongback	13											0.5		13			
PFIP support structure	14														14		
PFIP upper hexapod	15															15	0.2
PFIP Rho stage	16														0.3	0.3	16

Figure A6: Rework effort impact is the extent of redesign that would be required in the event of changes to a component with shared dependency.

The simulation output is a histogram representing a normal distribution of possible design effort cycle times.

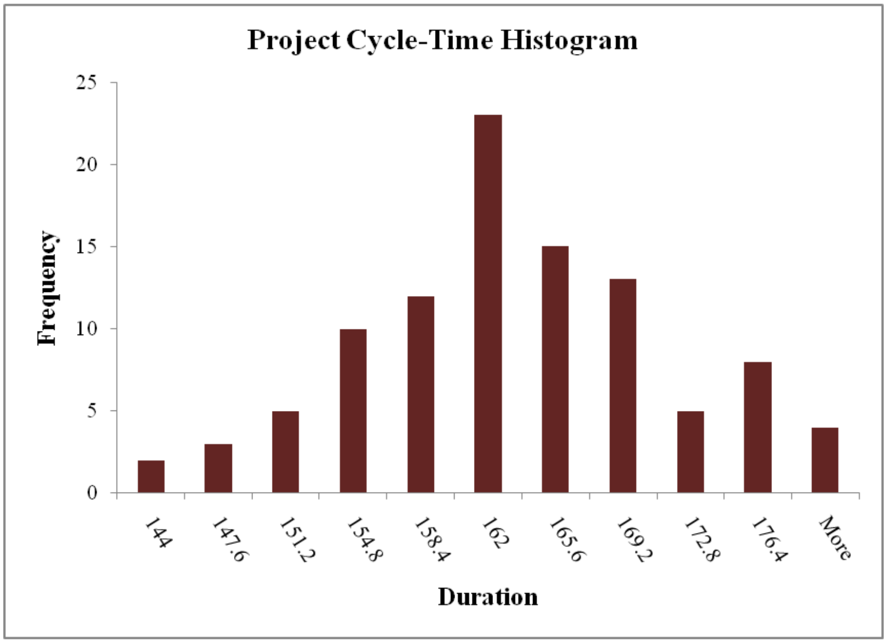

Figure A7: A frequency plot of the Monte-Carlo simulation output, representing a normal distribution of the estimated design project duration.

The DSM analysis methods above represent only a few of the capabilities currently available. There are Commercial DSM software packages which enhance these analysis capabilities, as well as build upon them to incorporate Multi-Domain-Mapping tools that span technical and social domains for additional benefits to design coordination, complexity management, and collaborative project planning.

References

[1] Bahill, A., and Botta, R., 2008, “Fundamental Principles of Good System Design,” Engineering Management Journal, Vol. 20, Iss. 4, pp. 9-17.

[2] Booth, J., et al., 2006, “The Wide Field Upgrade for the Hobby-Eberly Telescope,” *Proceedings of the 2006 SPIE Symposium on Ground-Based and Airborne Telescopes*, Vol. 6267.

[3] Conklin, J. 2005, *Dialogue Mapping: Building Shared Understanding of Wicked Problems*. Wiley, New York.

[4] Danilovic, M., and Sandkull, B., 2004, “The Use of Dependence Structure Matrix and Domain Mapping Matrix in Managing Uncertainty in Multiple Project Situations,” International Journal of Project Management, Vol. 23. pp. 193-203.

[5] Denker, S., Steward, D., and Browning, T., 2001, “Planning Concurrency and Managing Iteration in Projects,” Project Management Journal, Vol. 32, Iss. 3, pp. 31-38.

[6] Duffy, A., Ed., 1998, *The Design Productivity Debate*. Springer, Berlin.

[7] Frankenberger, E., Badke-Schaub, P., and Birkhofer, H., Eds., 1998, *Designers: The Key to Successful Product Development*. Springer, London.

[8] HETDEX: Leading the Revolution. Available at <http://hetdex.org/hetdex/index.php> (2009/6/14).

[9] “HETDEX Tracker and Fiber Management System Specification.” University of Texas Center for Electromechanics, Internal Doc. No. RF272, 2009.

[10] Hill, G., et al., “VIRUS: A Massively-Replicated Integral Field Spectrograph for HET,” *Proceedings of the 2006 SPIE Symposium on Ground-Based and Airborne Instrumentation for Astronomy*, Vol. 6269.

[11] Lindemann, U., 2009, The Design Structure Matrix. Available at <http://www.dsmweb.org> (2009/6/17).

[12] Lindemann, U., Maurer, M., and Braun, T., 2009, *Structural Complexity Management: An Approach for the Field of Product Design*. Springer, Berlin.

[13] Maheswari, J., and Varghese, K., 2005, “Project scheduling Using Dependency Structure Matrix,” International Journal of Project Management, Vol. 23, pp. 223-230.

[14] Mesihovic, S., Malmqvist, J., and Pikosz, P., 2004, “Product Data Management System-Based Support for Engineering Project Management,” Journal of Engineering Design, Vol. 15, Iss. 4, pp. 389-403.

[15] Oakley, M., Ed., 1990, *Design Management: A Handbook of Issues and Methods*. Basil Blackwell Ltd., Oxford.

[16] Reinertsen, D., 1997, *Managing the Design Factory*. The Free Press, New York.

[17] Schrage, M., 1989, *No More Teams! Mastering the Dynamics of Creative Collaboration*. Doubleday, New York.

[18] Suh, N., 2005, *Complexity: Theory and Applications.* Oxford University Press, New York.

Vita

Nicholas Taylor Mollison was born June 18, 1979 in Springdale, Arkansas to parents Richard A. Mollison and Cindy D. Mollison. After receiving his high school diploma from the Episcopal School of Dallas, Texas in May 1998, Nicholas attended The University of Texas at Austin. He received the degree of Bachelor of Science in Mechanical Engineering from the University of Texas in May 2003. Nicholas was employed by CRC-Evans Automatic Welding, Inc. in Houston, Texas as a product development engineer from August 2003 to June 2005. In July 2005, Nicholas returned to Austin, Texas where he was employed as a Research Engineer Associate at the University of Texas Center for Electromechanics.

Permanent address: 1203-B W. 39 ½ Street, Austin, Texas 78756

This report was typed by the author.